

(Mathematical) Logic for Systems Biology (Invited Paper)

Joëlle Despeyroux *

INRIA and CNRS, I3S, Sophia-Antipolis, France
joelle.despeyroux@inria.fr

Abstract. We advocate here the use of (mathematical) logic for systems biology, as a unified framework well suited for both modeling the dynamic behaviour of biological systems, expressing properties of them, and verifying these properties. The potential candidate logics should have a traditional proof theoretic pedigree (including a sequent calculus presentation enjoying cut-elimination and focusing), and should come with (certified) proof tools. Beyond providing a reliable framework, this allows the adequate encodings of our biological systems. We present two candidate logics (two modal extensions of linear logic, called HyLL and SELL), along with biological examples. The examples we have considered so far are very simple ones - coming with completely formal (interactive) proofs in Coq. Future works include using automatic provers, which would extend existing automatic provers for linear logic. This should enable us to specify and study more realistic examples in systems biology, biomedicine (diagnosis and prognosis), and eventually neuroscience.

1 Introduction

We consider here the question of reasoning about biological systems in (mathematical) logic. We show that two new logics, both modal extensions of linear logic [12] (LL), are particularly well-suited to this purpose. The first logic, called Hybrid Linear Logic (HyLL), has been developed by the author in joint work with K. Chaudhuri [8]. The second logic, an extension of Subexponential Linear Logic (SELL^m), has been independently proposed by C. Olarte, E. Pimentel and V. Nigam [15]. Both HyLL and SELL provide a unified framework to encode biological systems, to express temporal properties of their dynamic behaviour, and to prove these properties. By constructing proofs in the logics, we directly witness reachability as logical entailment [13, 17]. This approach is in contrast to most current approaches to applying formal methods to systems biology, which generally encode biological systems either in a dedicated programming language [6, 10, 19], or in differential equations [5], express properties in a temporal logic, and then verify these properties against some form of traces (model-checking), eventually built using an external simulator.

In a joint work with E. De Maria and A. Felty, we presented some first applications of HyLL to systems biology [13]. In these first experiments, we focused on

* CMSB 2016, Cambridge, UK. LNBI 9859. DOI 10.1007/978-3-319-45177-0

Boolean systems and in this case a time unit corresponds to a transition in the system. We believe that discrete modeling is crucial in systems biology because it allows taking into account some phenomena that have a very low chance of happening (and could thus be neglected by differential approaches), but which may have a strong impact on system behavior.

In a recent joint work with C. Olarte and E. Pimentel [9], we compared HyLL and SELL, providing two encodings. The first encoding is from HyLL’s logical rules into LL with the highest level of adequacy, hence showing that HyLL is as expressive as LL. We also proposed an encoding of HyLL into SELL^{m} , showing that SELL^{m} is more expressive than HyLL. However, the simplicity of HyLL might be of interest, both from the user point of view and as far as proof search is concerned (a priori easier and more efficient in HyLL than in SELL). In this joint work, we furthermore encoded temporal operators of Computational Tree Logic (CTL) into linear logic with fixed point operators.

We first recall here these two previous works. Then we briefly mention our current joint work with P. Lio, on formalizing the evolution of cancer cells, concluding with some future work.

This note is thus based on joint works with K. Chaudhuri (INRIA Saclay), A. Felty (Univ. of Ottawa), P. Lio (Cambridge Univ.), and C. Olarte and E. Pimentel (Universidade Federal do Rio Grande do Norte, Brazil).

2 Preliminaries

Although we assume that the reader is familiar with linear logic [12] (LL), we review some of its basic proof theory in the following sections. First, let us gently introduce linear logic by means of an example.

2.1 Linear Logic for Biology

Linear Logic (LL) [12] is particularly well suited for describing state transition systems. LL has been successfully used to model such diverse systems as: the π -calculus, concurrent ML, security protocols, multiset rewriting, and games.

In the area of biology, a rule of activation (e.g., a protein activates a gene or the transcription of another protein) can be modeled by the following LL axiom:

$$\text{active}(a, b) \stackrel{\text{def}}{=} \text{pres}(a) \multimap (\text{pres}(a) \otimes \text{pres}(b)).$$

The formula $\text{active}(a, b)$ describes the fact that a state where a is present ($\text{pres}(a)$ is true) can evolve into a state where both $\text{pres}(a)$ and $\text{pres}(b)$ are true.

Propositions such as $\text{pres}(a)$ are called *resources*, and a rule in the logic can be viewed as a rewrite rule from a set of resources into another set of resources, where a set of resources describes a state of the system. Thus, a particular state transition system can be modeled by a set of rules of the above shape. The rules of the logic then allow us to prove some desired properties of the system, such as, for

example, the existence of a stable state. However, linear implication is timeless. Linear implication can be used to model one event occurring after another, but it cannot be precise about how many steps or how long the delay is without explicitly encoding time. In a domain where resources have lifetimes and state changes have temporal, probabilistic or stochastic *constraints*, then the logic will allow inferences that may not be realizable in the system being modeled. This was the motivation of the development of HyLL, which was designed to represent constrained transition systems.

2.2 Linear Logic and Focusing

Literals are either atomic formulas (p) or their negations (p^\perp). The connectives \otimes and \wp and their units 1 and \perp are *multiplicative*; the connectives \oplus and $\&$ and their units 0 and \top are *additive*; \forall and \exists are (first-order) quantifiers; and $!$ and $?$ are the exponentials (called bang and question-mark, respectively).

First proposed by Andreoli [1] for linear logic, focused proof systems provide normal form proofs for cut-free proofs. The connectives of linear logic can be divided into two classes. The *negative* connectives have invertible introduction rules: these connectives are \wp , \perp , $\&$, \top , \forall , and $?$. The *positive* connectives \otimes , 1 , \oplus , 0 , \exists , and $!$ are the de Morgan duals of the negative connectives. A formula is *positive* if it is a negated atom or its top-level logical connective is positive. Similarly, a formula is *negative* if it is an atom or its top-level logical connective is negative.

Focused proofs are organized into two *phases*. In the *negative* phase, all the invertible inference rules are eagerly applied. The *positive* phase begins by choosing a positive formula F on which to focus. Positive rules are applied to F until either 1 or a negated atom is encountered (and the proof must end by applying the initial rules), the promotion rule ($!$) is applied, or a negative subformula is encountered and the proof switches to the negative phase.

This change of phases on proof search is particularly interesting when the focused formula is a *bipole* [1]. Focusing on a bipole will produce a single positive and a single negative phase. This two-phase decomposition enables us to adequately capture the application of object-level inference rules by the meta-level linear logic, as shown in [9].

2.3 Hybrid Linear Logic

Hybrid Linear Logic (HyLL) is a conservative extension of Intuitionistic first-order Linear Logic (ILL) [12] where the truth judgments are labelled by worlds representing constraints on states and state transitions. Instead of the ordinary judgment “ A is true”, for a proposition A , judgments of HyLL are of the form “ A is true at world w ”, abbreviated as $A @ w$. Particular choices of worlds produce particular instances of HyLL. Typical examples are “ A is true at time t ”, or “ A is true with probability p ”. HyLL was first proposed in [8] and it has been used as a logical framework for specifying biological systems [13].

Formally, worlds are defined as follows.

Definition 1 (HyLL worlds). A constraint domain \mathcal{W} is a monoid structure $\langle W, \cdot, \iota \rangle$. The elements of W are called worlds and its reachability relation $\preceq : W \times W$ is defined as $u \preceq w$ if there exists $v \in W$ such that $u \cdot v = w$.

The identity world ι is \preceq -initial and is intended to represent the lack of any constraints. Thus, the ordinary first-order linear logic is embeddable into any instance of HyLL by setting all world labels to the identity. A typical, simple example of constraint domain is $\mathcal{T} = \langle \mathbb{N}, +, 0 \rangle$, representing instants of time.

Atomic propositions (p, q, \dots) are applied to a sequence of terms (s, t, \dots) , which are drawn from an untyped term language containing constants (c, d, \dots) , term variables (x, y, \dots) and function symbols (f, g, \dots) applied to a list of terms (\mathbf{t}) . Non-atomic propositions are constructed from the connectives of first-order intuitionistic linear logic and the two hybrid connectives *satisfaction* (\mathbf{at}), which states that a proposition is true at a given world $(w, \iota, u \cdot v, \dots)$, and *localization* (\downarrow), which binds a name for the (current) world the proposition is true at. The following grammar summarizes the syntax of HyLL.

$$\begin{aligned} t &::= c \mid x \mid f(\mathbf{t}) \\ A, B &::= p(\mathbf{t}) \mid A \otimes B \mid \mathbf{1} \mid A \rightarrow B \mid A \& B \mid \top \mid A \oplus B \mid \mathbf{0} \mid !A \mid \\ &\quad \forall x. A \mid \exists x. A \mid (A \mathbf{at} w) \mid \downarrow u. A \mid \forall u. A \mid \exists u. A \end{aligned}$$

Note that world u is bounded in the propositions $\downarrow u. A$, $\forall u. A$ and $\exists u. A$. World variables cannot be used in terms, and neither can term variables occur in worlds. This restriction is important for the modular design of HyLL because it keeps purely logical truth separate from constraint truth. We note that \downarrow and \mathbf{at} commute freely with all non-hybrid connectives [8].

The sequent calculus [11] presentation of HyLL uses sequents of the form $\Gamma; \Delta \vdash C @ w$ where Γ (*unbounded context*) is a set and Δ (*linear context*) is a multiset of judgments of the form $A @ w$. Note that in a judgment $A @ w$ (as in a proposition $A \mathbf{at} w$), w can be any expression in \mathcal{W} , not only a variable.

The inference rules dealing with the new hybrid connectives are depicted below (the complete set of rules can be found in [8]).

$$\begin{array}{c} \frac{\Gamma; \Delta \vdash A @ u}{\Gamma; \Delta \vdash (A \mathbf{at} u) @ w} \mathbf{at}R \qquad \frac{\Gamma; \Delta, A @ u \vdash C @ w}{\Gamma; \Delta, (A \mathbf{at} u) @ v \vdash C @ w} \mathbf{at}L \\[10pt] \frac{\Gamma; \Delta \vdash A[w/u] @ w}{\Gamma; \Delta \vdash \downarrow u. A @ w} \downarrow R \qquad \frac{\Gamma; \Delta, A[v/u] @ v \vdash C @ w}{\Gamma; \Delta, \downarrow u. A @ v \vdash C @ w} \downarrow L \end{array}$$

Note that $(A \mathbf{at} u)$ is a *mobile* proposition: it carries with it the world at which it is true. Weakening and contraction are admissible rules for the unbounded context.

The most important structural properties are the admissibility of the general identity (i.e. over any formulas, not only atomic propositions) and cut theorems. While the first provides a syntactic completeness theorem for the logic, the latter guarantees consistency (i.e. that there is no proof of $\cdot \vdash \mathbf{0} @ w$).

Theorem 1 (Identity/Cut).

1. $\Gamma; A @ w \vdash A @ w$
2. If $\Gamma; \Delta \vdash A @ u$ and $\Gamma; \Delta', A @ u \vdash C @ w$, then $\Gamma; \Delta, \Delta' \vdash C @ w$
3. If $\Gamma; \cdot \vdash A @ u$ and $\Gamma, A @ u; \Delta \vdash C @ w$, then $\Gamma; \Delta \vdash C @ w$.

Moreover, HyLL is conservative with respect to intuitionistic linear logic: as long as no hybrid connectives are used, the proofs in HyLL are identical to those in ILL. It is worth noting that HyLL is more expressive than S5, as it allows direct manipulation of the worlds using the hybrid connectives and HyLL's δ connective (see Section 5) is not definable in S5. We also note that HyLL admits a complete focused [1] proof system. The interested reader can find proofs and further meta-theoretical theorems about HyLL in [8].

Modal Connectives. We can define modal connectives in HyLL as follows:

Definition 2 (Modal connectives).

$$\Box A \stackrel{\text{def}}{=} \downarrow u. \forall w. (A \text{ at } u.w) \quad \Diamond A \stackrel{\text{def}}{=} \downarrow u. \exists w. (A \text{ at } u.w) \quad \delta_v A \stackrel{\text{def}}{=} \downarrow u. (A \text{ at } u.v)$$

$\Box A$ [resp. $\Diamond A$] represents all [resp. some] state(s) satisfying A and reachable from now. The connective δ represents a form of delay.

2.4 Subexponentials in Linear Logic

Linear logic with subexponentials (SELL) shares with LL all its connectives except the exponentials: instead of having a single pair of exponentials $!$ and $?$, SELL may contain as many *subexponentials* [7, 18], written $!^a$ and $?^a$, as one needs. The grammar of formulas in SELL is as follows:

$$F ::= \mathbf{0} \mid 1 \mid \top \mid \perp \mid p(\mathbf{t}) \mid F_1 \otimes F_2 \mid F_1 \oplus F_2 \mid F_1 \wp F_2 \mid F_1 \& F_2 \mid \\ \exists x.F \mid \forall x.F \mid !^a F \mid ?^a F$$

The proof system for SELL is specified by a *subexponential signature* $\Sigma = \langle I, \preceq, U \rangle$, where I is a set of labels, $U \subseteq I$ is a set specifying which subexponentials allow weakening and contraction, and \preceq is a pre-order among the elements of I . We shall use a, b, \dots to range over elements in I and we will assume that \preceq is upwardly closed with respect to U , i.e., if $a \in U$ and $a \preceq b$, then $b \in U$.

The system SELL is constructed by adding all the rules for the linear logic connectives except for the exponentials. The rules for subexponentials are dereliction and promotion of the subexponential labelled with $a \in I$

$$\frac{\vdash ?^{a_1} F_1, \dots, ?^{a_n} F_n, G}{\vdash ?^{a_1} F_1, \dots, ?^{a_n} F_n, !^a G} !^a \quad \frac{\vdash \Gamma, G}{\vdash \Gamma, ?^a G} ?^a$$

Here, the rule $!^a$ has the side condition that $a \preceq a_i$ for all i . That is, one can only introduce a $!^a$ on the right if all other formulas in the sequent are marked with indices that are greater or equal than a . Moreover, for all indices $a \in U$, we add the usual rules for weakening and contraction.

We can enhance the expressiveness of SELL with the subexponential quantifiers \mathbb{M} and \mathbb{U} ([15, 18]) given by the rules (omitting the subexponential signature)

$$\frac{\vdash \Gamma, G[l_e/l_x]}{\vdash \Gamma, \mathbb{M}l_x : a.G} \mathbb{M} \quad \frac{\vdash \Gamma, G[l/l_x]}{\vdash \Gamma, \mathbb{U}l_x : a.G} \mathbb{U}$$

where l_e is fresh. Intuitively, subexponential variables play a similar role as eigen-variables. The generic variable $l_x : a$ represents any subexponential, constant or variable in the ideal of a . Hence l_x can be substituted by any subexponential l of type b (i.e., $l : b$) if $b \preceq a$. We call the resulting system $\text{SELL}^\mathfrak{M}$.

As shown in [15, 18], $\text{SELL}^\mathfrak{M}$ admits a cut-free and also a complete focused proof system.

Theorem 2. *$\text{SELL}^\mathfrak{M}$ admits cut-elimination for any subexponential signature.*

Modal connectives. We can define modal connectives in SELL as follows:

$$\Box_u A \stackrel{\text{def}}{=} \forall l : u. !^l A \quad \Diamond_u A \stackrel{\text{def}}{=} \exists l : u. !^l A \quad \Box A \stackrel{\text{def}}{=} \forall t : \infty. !^t A \quad \Diamond A \stackrel{\text{def}}{=} \exists t : \infty. !^t A$$

3 First experiments with HyLL

In a joint work with E. De Maria and A. Felty, we presented some first applications of HyLL to systems biology [13]. In these first experiments, we focused on Boolean systems and in this case a time unit corresponds to a transition in the system.

The activation rule seen in LL (Sec. 2.1) can be written in HyLL as

$$\text{active}(a, b) \stackrel{\text{def}}{=} \text{pres}(a) \multimap \delta_1 (\text{pres}(a) \otimes \text{pres}(b)).$$

We chose a simple yet representative biological example concerning the DNA-damage repair mechanism based on proteins p53 and Mdm2, and present and proved several properties of this system. All these properties were reachability properties or the existence of an invariant. Most interesting proofs require induction or case analysis, that we borrowed from the meta-level (Coq). We fully formalized these proofs in the Coq Proof Assistant [3]. In Coq, we can both reason in HyLL and formalize meta-theoretic properties about it.

We discussed the merits and eventual drawbacks of this new approach compared to approaches using temporal logic and model checking. To better illustrate the correspondence with such approaches, which all use temporal logic to reason about (simulations of models of) the biological systems described, we also presented, informally but in some detail, the encoding of temporal logic operators in HyLL.

4 Relative Expressiveness Power of HyLL and SELL

We observe that, while linear logic has only seven logically distinct prefixes of bangs and question-marks, SELL allows for an unbounded number of such prefixes, e.g., $!^i$, or $!^{i?j}$. Hence, by using different prefixes, we allow for the specification of richer systems where subexponentials are used to mark different modalities/states. For instance, subexponentials can be used to represent contexts of proof systems [16]; to specify systems with temporal, epistemic and

spatial modalities [18] and to specify and verify biological systems [17]. An inhibition rule can be written in (classical) SELL as

$$\mathbf{inhib}(a, b) \stackrel{\text{def}}{=} !^t a \multimap !^{t+1}(a \otimes b^\perp).$$

HyLL and Linear Logic. One may wonder whether the use of worlds in HyLL increases also the expressiveness of LL. In a joint work with C. Olarte and E. Pimentel [9], we proved that this is not the case, by showing that HyLL rules can be directly encoded into LL by using the methods proposed in [14]. Moreover, the encoding of HyLL into LL is *adequate* in the sense that a focused step in LL corresponds *exactly* to the application of one inference rule in HyLL.

HyLL and SELL. Linear logic allows for the specification of two kinds of context maintenance: both weakening and contraction are available (classical context) or neither is available (linear context). That is, when we encode (linear) judgments in HyLL belonging to different worlds, the resulting meta-level atomic formulas will be stored in the same (linear) LL context. The same happens with classical HyLL judgments and the classical LL context.

Although this is perfectly fine, encoding HyLL into $\text{SELL}^{\mathfrak{m}}$ allows for a better understanding of worlds in HyLL. For that, we use subexponentials to represent worlds, having each world as a linear context. A HyLL judgment of the shape $F@w$ in the (left) linear context is encoded as the $\text{SELL}^{\mathfrak{m}}$ formula $?^w[F@w]$. Hence, HyLL judgments that hold at world w are stored at the w linear context of $\text{SELL}^{\mathfrak{m}}$. A judgment of the form $G@w$ in the classical HyLL context is encoded as the $\text{SELL}^{\mathfrak{m}}$ formula $?^c ?^w[G@w]$. Then, the encoding of $G@w$ is stored in the unbounded (classical) subexponential context c .

We showed that our encoding is indeed adequate. Moreover, as before, the adequacy of the encodings is on the *level of derivations*.

Information Confinement. One of the features needed to specify spatial modalities is information *confinement*: a space/world can be inconsistent and this does not imply the inconsistency of the whole system. We showed in [9] that information confinement cannot be specified in HyLL. The authors in [15] exploit the combination of subexponentials of the form $!^w ?^w$ in order to specify information confinement in $\text{SELL}^{\mathfrak{m}}$. More precisely, note that the sequents (in a 2-sided presentation of SELL) $!^w ?^w \mathbf{0} \not\vdash \mathbf{0}$ and $!^w ?^w \mathbf{0} \not\vdash !^v ?^v \mathbf{0}$, representing “inconsistency is local” and “inconsistency is not propagated” respectively hold in SELL.

5 Computation Tree Logic (CTL) in Linear Logic.

Hybrid linear logic is expressive enough to encode some forms of modal operators, thus allowing for the specification of properties of transition systems. As mentioned in [13], it is possible to encode CTL temporal operators into HyLL considering existential (E) and bounded universal (A) path quantifiers. We extended these encodings in [9], showing how to fully capture E and A CTL quantifiers in linear logic with fixed points. For that, we used the system μMALL [2]

that extends MALL (multiplicative, additive linear logic) with fixed point operators. In [13], proofs of (encodings of) properties involving CTL quantifiers use induction borrowed from the (Coq) meta-level. In [9], we could directly use fixed points in linear logic.

6 Concluding Remarks and Future Work

Concerning related work, it is worth noticing that there are some other logical frameworks that are extensions of LL, for example, HLF [20]. Being a logic in the LF family, HLF is based on natural deduction, hence having a complex notion of $(\beta\eta)$ normal forms. Thus adequacy (of encodings of systems) results are often much harder to prove in HLF than in (focused) HyLL/SELL. HLF seems to have been later abandoned in favour of Hybridized Intuitionistic Linear Logic (HILL) [4] - a type theory based on a subpart of HyLL.

Both HyLL and SELL have been used for formalizing and analyzing biological systems [13, 17]. SELL proved to be a broader framework for handling such systems (in particular localities). However, the simplicity of HyLL may be of interest for specific purposes, such as building tools for diagnosis in biomedicine.

Formal proofs in HyLL were implemented in [13], in the Coq [3] proof assistant. It would be interesting to extend the implementations of HyLL given there to SELL. Such an interactive proof environment would enable both formal studies of encoded systems in SELL and formal meta-theoretical study of SELL itself.

We may pursue the goal of using HyLL/SELL for further applications. That might include neuroscience, a young and promising science where many hypotheses are provided and need to be verified. Indeed, logic is a general tool whose area of potential applications are not restricted per se. This is in contrast to most of the other approaches, which are valid only in a restricted area (typically inside or outside the cell).

In an ongoing joint work with P. Lio, we are formalizing the evolution of cancer cells, acquiring driver or passenger mutations. A rule describing an intravasating Circulating Tumour Cell, for example, might be:

$$\mathcal{C}(n, \textit{breast}, f, [\textit{EPCAM}]) \multimap_d \mathcal{C}(n, \textit{blood}, 1, [\textit{EPCAM}])$$

where f is a fitness parameter, here in $\{0, 1\}$. Our long term goal here is the design of a Logical Framework for disease diagnosis and therapy prognosis. This requires the development of automatic tools for proof search in our logics. These tools should benefit both from current research on proof search in linear logic and from current developments of automatic provers for SELL.

References

1. Andreoli, J.: Logic programming with focusing proofs in linear logic. J. Log. Comput. 2(3), 297–347 (1992)

2. Baelde, D.: Least and greatest fixed points in linear logic. *ACM Trans. Comput. Log.* 13(1), 2 (2012)
3. Bertot, Y., Castéran, P.: *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions*. Springer (2004)
4. Caires, L., Perez, J., Pfenning, F.: Logic-based domain-aware session types (2014), submitted.
5. Campagna, D., Piazza, C.: Hybrid automata in systems biology: How far can we go? *Electronic Notes in TCS* 229(1), 93–108 (2009)
6. Danos, V.: Agile modelling of cellular signalling (invited paper). In: *Proceedings of the 5th Workshop on Structural Operational Semantics (SOS)*. *Electronic Notes in TCS*, vol. 229, pp. 3–10. Elsevier (2009)
7. Danos, V., Joinet, J.B., Schellinx, H.: The structure of exponentials: Uncovering the dynamics of linear logic proofs. In: Gottlob, G., Leitsch, A., Mundici, D. (eds.) *Kurt Gödel Colloquium. LNCS*, vol. 713, pp. 159–171. Springer (1993)
8. Despeyroux, J., Chaudhuri, K.: A hybrid linear logic for constrained transition systems. In: *Post-Proceedings of the 9th Intl. Conference on Types for Proofs and Programs (TYPES 2013)*. *Leibniz Intl. Proceedings in Informatics*, vol. 26, pp. 150–168. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2014)
9. Despeyroux, J., Olarte, C., Pimentel, E.: Hybrid and subexponential linear logics. In: *Proceedings of the 11th workshop on Logical and Semantic Frameworks, with Applications (LSFA)* (2016)
10. Fages, F., Soliman, S.: Formal cell biology in BIOCHAM. In: Bernardo, M., Degano, P., Zavattaro, G. (eds.) *8th Intl. School on Formal Methods for the Design of Computer, Communication and Software Systems: Computational Systems Biology SFM'08. LNCS*, vol. 5016, pp. 54–80. Springer-Verlag, Bertinoro, Italy (Feb 2008)
11. Gentzen, G.: Investigations into logical deductions, 1935. In: Szabo, M.E. (ed.) *The Collected Papers of Gerhard Gentzen*, pp. 68–131. North-Holland Publishing Co., Amsterdam (1969)
12. Girard, J.Y.: Linear logic. *TCS* 50, 1–102 (1987)
13. de Maria, E., Despeyroux, J., Felty, A.: A logical framework for systems biology. In: *Proceedings of the 1st Intl. Conference on Formal Methods in Macro-Biology (FMMB)*. *LNCS*, vol. 8738, pp. 136–155. Springer (2014)
14. Miller, D., Pimentel, E.: A formal framework for specifying sequent calculus proof systems. *Theor. Comput. Sci.* 474, 98–116 (2013)
15. Nigam, V., Olarte, C., Pimentel, E.: A general proof system for modalities in concurrent constraint programming. In: *Proceedings of the 24th intl. conference on Concurrency theory (CONCUR)*. *LNCS*, vol. 8052, pp. 410–424. Springer Verlag (2013)
16. Nigam, V., Pimentel, E., Reis, G.: Specifying proof systems in linear logic with subexponentials. *Electr. Notes Theor. Comput. Sci.* 269, 109–123 (2011)
17. Olarte, C., Chiarugi, D., Falaschi, M., Hermith, D.: A proof theoretic view of spatial and temporal dependencies in biochemical systems. *TCS* 641, 25–42 (2016)
18. Olarte, C., Pimentel, E., Nigam, V.: Subexponential concurrent constraint programming. *TCS* 606, 98–120 (2015)
19. Phillips, A., Cardelli, L.: A correct abstract machine for the stochastic pi-calculus. In: *BioConcur: Workshop on Concurrent Models in Molecular Biology*. *Electronic Notes in TCS* (2004)
20. Reed, J.: Hybridizing a logical framework. In: *International Workshop on Hybrid Logic (HyLo)*. Seattle, USA (Aug 2006)